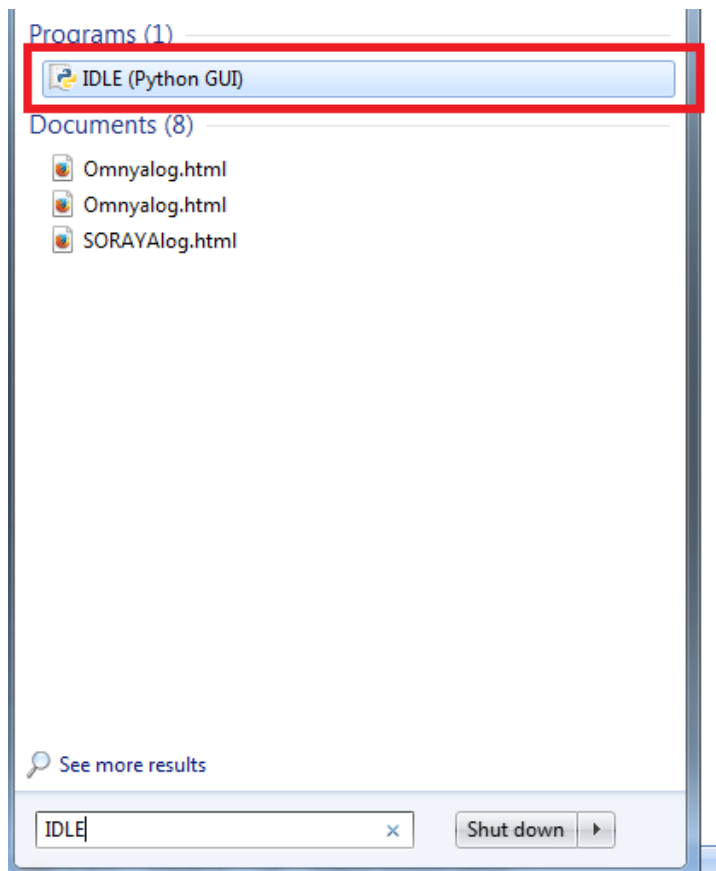


Help In Python

--> open python IDLE



--> to search help for Module

just import this module --> then write `help(module)`

```
>>> import math
>>> help(math)
Help on built-in module math:

NAME
    math

FILE
    (built-in)

DESCRIPTION
    This module is always available. It provides access to the
    mathematical functions defined by the C standard.

FUNCTIONS
    acos(...)
        acos(x)

        Return the arc cosine (measured in radians) of x.

    acosh(...)
        acosh(x)

        Return the inverse hyperbolic cosine of x.

    asin(...)
        asin(x)

        Return the arc sine (measured in radians) of x.
```

also

```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> help(cv2)
Help on module cv2:

NAME
    cv2

FILE
    c:\python27\lib\site-packages\cv2.pyd

SUBMODULES
    Error
    detail
    fisheye
    flann
    instr
    ml
    ocl
    ogl
    videostab

CLASSES
    __builtin__.object
        UMat
    exceptions.Exception(exceptions.BaseException)
        error

    class UMat(__builtin__.object)
        | OpenCV 3 UMat wrapper. Used for T-API support.
        |
        | Methods defined here:
        |
        | __init__(...)
        |     x.__init__(...) initializes x; see help(type(x)) for signature
        |
        | get(...)
        |     Returns numpy array
```

--> to search for specific method or property in a module

1-import module

2-help(module.method)

```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> help(cv2.Canny)
Help on built-in function Canny:

Canny(...)
    Canny(image, threshold1, threshold2[, edges[, apertureSize[, L2gradient]]])
-> edges or Canny(dx, dy, threshold1, threshold2[, edges[, L2gradient]]) -> edges

>>> |
```